REMARKS

The above amendments to the above-captioned application along with the following remarks are being submitted as a full and complete response to the Final Official Action dated December 12, 2004. In view of the above amendments and the following remarks, the Examiner is respectfully requested to give due reconsideration to this application, to indicate the allowability of the claims, and to pass this case to issue.

Status of the Claims

Claims 1-8 currently stand and are under consideration in this application. Applicants hereby submit that no new matter is being introduced into the application through the submission of this response.

Prior Art Rejections

Claims 1 and 2 were rejected under 35 U.S.C. § 102(b) on the grounds of being anticipated by a publication of Intel Corp.: IA-64 application Developer's Architecture Guide, May 1999 (hereinafter "Intel"). Claim 3 was rejected under 35 U.S.C. § 102(e) on the grounds of being anticipated by U.S. Patent No. 6,463,525 to Prabhu. Further, claim 5 was rejected under 35 U.S.C. 103(a) on the grounds of being unpatentable over Intel. These rejections have been carefully considered, but are most respectfully traversed.

The processor of the invention, as now recited in claim 1, comprises: a register file including a plurality of registers assigned with register numbers, each of the registers storing operand data; a plurality of operation pipes, each operation pipe for executing in parallel one kind of operation on a plurality of the operand data; a decoder for decoding a register designating field of an instruction code, the register designating field having a register number stored therewith, the decoder further for generating signals designating register numbers based on the register number of the register designating field, the designated register numbers being consecutive with the register number of the register designating field; and a control circuit for sending operand data stored in the registers corresponding to the designated register numbers to at least one of the operation pipes such that the at least one of the operation pipes executes in parallel the one kind of operation associated therewith on the operand data sent from the corresponding designated registers pipes for executing the operation therein in parallel.

The present invention as recited in claim 2 is directed to a processor comprising: a register file including a plurality of registers assigned with register numbers; a plurality of operation pipes, each operation pipe for executing in parallel one kind of operation on a plurality of the operand data so as to generate operation result data; a decoder for decoding a register designating field of an instruction code, the register designating field having a register number stored therewith, the decoder further for generating signals designating register numbers based on and consecutive with the register number of the register designating field, and the registers corresponding to the designated register numbers being designated for storing the operation result data; and a control circuit for sending the operation result data from at least one of the operation pipes to the corresponding designated registers.

As recited in claim 3, the present invention is directed to a processor comprising: a register file including a plurality of registers assigned with register numbers, each of the registers for storing at least one of operand data and operation result data; a plurality of operation pipes, each operation pipe for executing in parallel one kind of operation on a plurality of the operand data so as to generate the operation result data; a first decoder for decoding a first register designating field of an instruction code, the first register designating field having a first register number stored therewith, the first decoder further for generating signals designating source register numbers based on and consecutive with the first register number; a second decoder for decoding a second register designating field of the instruction code, the second register designating field having a second register number stored therewith, the second decoder further for generating signals designating result register numbers based on and consecutive with the second register number; and a control circuit for sending the operand data stored in source registers corresponding to the designated source register numbers to at least one of the operation pipes such that the at least one of the operation pipes executes in parallel the one kind of operation associated therewith on the operand data and for sending the operation result data obtained from the at least one operation pipe to result registers corresponding to the designated result register numbers.

With respect to the Examiner's argument regarding "said decoder further for generating signals designating register numbers based on the register number of the register field" in the present invention, Applicants understand that the Examiner has taken the position that this feature is anticipated by Intel's page 7-137, padd, page C-17 where signals designating register numbers r2 and r3 are generated. However, Intel in actuality only teaches that the signals can designate only one register by one register designating field in an

instruction code (See page C-17). The rl designating field causes a signal designating only rl, r2 designating field causes a signal designating only r2, and so on. Intel's page 7-137, padd, which uses rl, r2 and r3, is nothing more than an arithmetic operation based on this instruction decoding technique.

With respect to the Examiner's position on "said designated register numbers being consecutive with the register number of the register designating field", Applicants understand that the Examiner interprets Intel's r2 to be consecutive with r3. However, Applicants will point out that this is not accurate because r2 and r3 merely show the register notations in an instruction format, with each representing a register independent each other, so that r2 as a register number is not consecutive with r3 in general. If r2 as a register number must be consecutive with r3, this would impose great restrictions on the use of registers in programming that would contradict the intent and purpose of the Intel reference, as well as the understanding of the use of registers as would be known to those of skill in the art.

On the other hand, the present invention allows one register designating field to designate a plurality of registers. For example, in the case that an instruction code has 3 register designating fields like Intel's IA-64, supposing that one register designating field allows to designate 4 registers, the first register designating field allows to designate R1, R2, R3 and R4 registers; the second field to designate R5, R6, R7 and R8 registers; and the third field to designate R9, R10, R11 and R12 registers. The register numbers R1-R4, R5-R8 and R9-R12 are consecutive within each register group.

In view of the above, one clear difference between the present invention and the reference is that Intel's register designating field allows only one direct register to be designated, while the equivalent field in the present invention allows more than one registers. This difference is attributed to the difference in processor architecture, and thus Applicant will contend that Intel cannot and does not anticipate or render obvious all the features of the present invention.

With respect to claim 2, Applicants understand that the Examiner has taken the position that "said decoder further for generating signals designating register numbers based on and consecutive with the register number of the register designating field, and the registers corresponding to the designated register numbers being designated for storing the operation result data" is anticipated by Intel's page 7-137, padd, page C-17 where signals designating register numbers r2 and r3 are generated. Here again, Applicants will point out that Intel's r2 and r3 are not consecutive register numbers in general. Intel's r2 designates directly one

independent register, while r3 designates another independent register.

In contrast, in the present invention, decoding one designating field can result in generating signals designating a plurality of consecutive registers such as R1, R2, R3 and R4 for storing the operation result data. Therefore, here again, Applicants will submit that the Intel reference cannot anticipate or render obvious every feature of the present invention as claimed.

Regarding claim 3, Applicants will point out that Prabhu allows a double precision operation by using single precision operands. Prabhu also designates only one register by one register designating field in a command code. Specifically, Prabhu's command is decoded in accordance with Prabhu's command sequence shown in TABLE 2 (similar in the case of TABLE 1) as follows:

(1) "load % **fo**" - Prabhu decodes a register designating field **fo** to designate a register with 32-bit operand to be contained (Prabhu, col. 4, lines 35-55). Similar in the commands "load % **f1**", "load % **f2**", and "load % **f3**".

(2) "merge % **fo**, **f1**→ **%dO**" - Prabhu decodes a register designating field **fo** to designate a register with 32-bit operand contained, field **f1** to designate another register with 32-bit operand contained, field **d0** to designate still another register with 64-bit operand to be contained. Note that **d0** specifies a 64-bit operand register (col. 4, line 35). **FO** and **f1** are supposed to be a pair of 32-bit operands used for realizing a software-implemented double precision operation, which utilizes an arithmetic operation hardware for single precision floating point operation. Similar in the command "merge % **f2**, **f3** → **%d2**"

(3) "fadd **%dO**,**%d2**,**%d4**" - Prabhu decodes a register designating field **dO** to designate a register with 64-bit operand contained, field **d2** to designate another register with 64-bit operand contained, and field **d4** to designate still another register with 64-bit operand to be contained (col. 4, lines 34-38).

Applicants will point out that the designating fields **dO**, **d2** and **d4** each only designates one register containing a 64-bit operand, not two registers each containing an original 32-bit operand. The 64-bit operand has a different data format from that of 32-bit operand (col. 1, lines 21-27). Further, **dO** in FIG. 2 is a 64-bit operand consisting of two converted (called "aliased") 32-bit operands into a 64-bit operand format.

Regarding the feature of "said first decoder further for generating signals designating

source register numbers based on and consecutive with the first register number" of the present invention, Applicants understand that the Examiner believes that this feature is anticipated by Prabhu's Figure 2, column 5, TABLE 2, where when **dO** is decoded, signals for **fo** and **fl** are generated. However, Applicants will contend that it is not correct to state that when **dO** is decoded, signals for **fO** and **fl** are generated, because **dO** does not imply a combination of **f0** and **fl**, but a single 64-bit operand originated from **fO** and **fl**. Prabhu in fact reveals that it only teaches designating or decoding only one register by one register designating field. Consequently, Applicants will contend that Prabhu also cannot anticipate or render obvious all the features of the present invention as claimed.

Accordingly, the present invention as now recited in the independent claims 1-3 is distinguishable and thereby allowable over the rejections raised in the Office Action. Further, since claim 5 is dependent on claim 1, the reference of Intel cannot also be used to render obvious additional features in a dependent claim when it could not anticipate or render obvious all the features in the parent independent claim. In other words, the present invention as recited in claim 5 is also distinguishable and thereby allowable over the rejections raised in the Office Action. The withdrawal of the outstanding prior art rejections is in order, and is respectfully solicited.

Allowable Subject Matter

Applicants thank the Examiner for his consideration in allowing claims 4 and 6 – 8. However, in view of the above arguments, Applicants will submit that all the claims are now allowable and thus no further amendments to the claims are now necessary.

Conclusion

In view of all the above, clear and distinct differences as discussed exist between the present invention as now claimed and the prior art reference upon which the rejections in the Office Action rely, Applicant respectfully contends that the prior art references cannot anticipate the present invention or render the present invention obvious. Rather, the present invention as a whole is distinguishable, and thereby allowable over the prior art.

Favorable reconsideration of this application is respectfully solicited. Should there be any outstanding issues requiring discussion that would further the prosecution and allowance of the above-captioned application, the Examiner is invited to contact the Applicant's undersigned representative at the address and phone number indicated below.

Respectfully submitted,

Stanley P. Fisher
Registration Number 24,344

Juan Carlos A. Marquez
Registration Number 34,072

REED SMITH LLP
3110 Fairview Park Drive, Suite 1400
Falls Church, Virginia 22042
(703) 641-4200

March 14, 2005

SPF/JCM/